SEMESTER 2 EXAMINATION – 2005/2006

SOLUTIONS

FIRST EXAMINATION IN ENGINEERING

Programme Codes: ENBDF0002, ENBDF0003, ENBDF0004, ENBDF0005, ENBDF0008, ENBDF0011

COMP 10060 Computer Science for Engineers

Prof. J. Kramer
Prof. B. Smyth
Dr. J. Murphy*

Time allowed: 2 hours

Answer **Question 1** and **one** other Question.

Question 1 carries 60 marks; Questions 2 and 3 carry 40 marks.

This is a closed-book examination. No calculators allowed.

Loose Rough Work sheets are not to be distributed or used.

READ EACH QUESTION CAREFULLY.

Solutions

Question 1 [60 marks]

(a) False (comments are optional in C)

(b) False (logic errors are not detected by the C compiler)

(c) value of w is 6

(d) value is 2.2361

(e) error case

(f) that one

(g) i is 7
i is 2
i is 0

(h) correct answer is (h-1) fname() takes 1 argument of type pointer-to-int and returns a value of type float.

(i) element number 1 is 1
element number 2 is 2
element number 3 is 3
element number 4 is 0
element number 5 is 0

(j) correct answer is (j-3) y = *ptr1;

(k) x is -1 and y is -2

(l) string=YbcdYfghYjklmn

(m) correct answer is (m-2) if it exists, open datafile.txt for reading only, otherwise return an error.

(n) The most widely used tools for developing algorithms are flowcharts and __pseudo-code__ .

(o) default
2
3
default
default

(p) result is 19

(q) x[0] is 2.00

x[1] is 1.00
x[2] is 0.67
y[0] is 2
y[1] is 1
y[2] is 0

(r) str=I Hope To Do Well On This Exam.

(s) value is 2.20

(t) correct answer is (t-1) emp1.age = 47;

Question 2 [40 marks]

For either part of these solutions, there should be flow charts or pseudo code

(a)

```
/*************************************************************************
*/
/* Question 2 (a)                                                        */
/*                                                                       */
/* Write a program that uses a function convert() to determine the       */
/* equivalent number of hours, minutes, and seconds for a given time in  */
/* seconds. For example, 3,661 seconds is equivalent to 1 hour, 1 minute, and */
/* 1 second. Use the following function prototype:                       */
/*                                                                       */
/* void convert(int time, int *phrs, int *pmins, int *psecs);            */
/*                                                                       */
/* convert() should put the number of hours into the int pointed to by phrs,  */
/* the number of minutes into the int pointed to by pmins, and the number of  */
/* seconds into the int pointed to by psecs. All input and output in the  */
/* program are to be done in main().                                     */
/*************************************************************************
*/

// Pre-processor directives
#include <stdio.h>

// Function prototype
void convert(int time, int *phrs, int *pmins, int *psecs);

// main() function
void main()
{
  int time;              // int to hold the time in secs
  int hrs, mins, secs;   // ints to hold the converted time
```

```
  // ask the user for a time in seconds
  printf( "\nPlease enter a time in seconds: " );
  scanf( "%d", &time );

  // call our signs function
  convert( time, &hrs, &mins, &secs );

  // Print out the results
  printf( "\n%d seconds is equivalent to: ", time );
  printf( "\n\t%dhrs %dmins %dsecs\n\n", hrs, mins, secs );

  // program ends
}


// the stats function
void convert(int time, int *phrs, int *pmins, int *psecs)
{
  // find the number of full hours
  // remember that integer division always rounds down to the nearest int
  *phrs = time / (60*60);      // 60*60secs = 1 hour

  // and subtract this from time
  time -= (*phrs * 60 * 60);   // time = time - (no. secs in "*phrs" hours)

  // now find the number of full minutes remaining
  *pmins = time / 60;

  // and subtract this from time
  time -= (*pmins*60);

  // whatever is left over is the number of secs
  *psecs = time;

  //done
}
```

(b)

```
/*************************************************************************/
/*   Question 2 (b)                                                      */
/*                                                                       */
/*   Write a program to compute the received Doppler frequency of a      */
/*   radar echo for a target moving towards the transmitter at velocities */
/*   ranging from 10 m/s to 100 m/s in 10 m/s steps. The received Doppler */
/*   frequency Fr is given by (Fr – Ft)/Ft=2V/c, where V is the velocity  */
/*   in m/s, Ft=5.5 GHz the transmitted frequency and c=3 10^8 m/s       */
/*************************************************************************/

/*************************************************************************/
/* Pre-processor directives                                              */
/*************************************************************************/

#include <stdio.h>
#define CONSTANT 300000000

/*************************************************************************/
/* main function (entry point for the program)                           */
/*************************************************************************/

int main()
{
  /* 1- Declaration of variables*/
  double freq_received, freq_transmitted;
  float velocity, velocity_max;


          /* print out table headers*/
  printf("\t_____\n");
  printf("\tVelocity\tReceived freq.\n");
  printf("\t(m/s)\t\t(MHz)\n");
  printf("\t-------------------------------\n");


  velocity_max=100;
  freq_transmitted=5500;

  /* 3- Use for loop to calculate the frequency received */
  for (velocity=10; velocity<=velocity_max; velocity=velocity+10)
  {
    freq_received=(2*velocity*freq_transmitted/CONSTANT)+freq_transmitted;
    printf("\t%.2f\t\t %f \n", velocity, freq_received);
  }
  return(0);
}
```

Question 3 [40 marks]

(a)

```
int numneg(int array[], int size){
int i, k=0; /* i is a local loop counter */
for (i=0; i<size; i++){
                    if (array[i]<0){
                    k++;
                    }
}
return (k);
}
```

(b) LINE 1: scanf("%d", intarr+i);
LINE 2: total += *(intarr+i);
LINE 3: printf("sum of inputs is %d, first input was %d,
last input was %d\n", total, *intarr, *(intarr+3));

(c) (i) After executing this program, the file output.dat contains 0 2 6

(ii)

```
#include <stdio.h>
void main(void)
{
  FILE *fptr1, *fptr2;
  int inp;
  fptr1 = fopen("input.dat", "r");
  fptr2 = fopen("output2.dat", "w");
  while (fscanf(fptr1, "%d", &inp)==1){
    if (inp >= -1 && inp <= 4){
      fprintf(fptr2, "%d ", inp);
    }
}
fclose(fptr1);
  fclose(fptr2);
 }
```

o O o